

## AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

### Listing of Claims:

1           1. (Currently amended) A method that predicts a result produced by a  
2 section of code in order to support speculative program execution, the section of  
3 code including a plurality of program instructions, the method comprising:  
4           executing the section of code within a program using a head thread,  
5 wherein executing the section of code produces the result;  
6           before the head thread produces the result, generating a predicted result to  
7 be used in place of the result;  
8           ~~allowing the speculative~~ allowing a speculative thread to speculatively execute  
9 subsequent code within the program using the predicted result, wherein the  
10 subsequent code follows the section of code in an execution stream of the  
11 program, and wherein speculatively executing the subsequent code involves  
12 performing one of:  
13                     a speculative method invocation to speculatively execute  
14                     the subsequent code,  
15                     a speculative function call to speculatively execute the  
16                     subsequent code, and  
17                     a speculative procedure call to speculatively execute the  
18                     subsequent code;  
19           wherein the head thread and all speculative threads execute instructions  
20 from separate instruction caches; and

A1  
21 after the head thread finishes executing the section of code, determining if  
22 a difference between the predicted result and the result generated by the head  
23 thread affected execution of the speculative thread;  
24 if the difference affected execution of the speculative thread, executing the  
25 subsequent code again using the result generated by the head thread; and  
26 if the difference did not affect execution of the speculative thread,  
27 performing a join operation to merge state associated with the speculative thread  
28 with state associated with the head thread.

1 2. (Original) The method of claim 1, wherein executing the subsequent  
2 code again involves performing a rollback operation for the speculative thread to  
3 undo actions performed by the speculative thread.

1 3. (Original) The method of claim 1, wherein determining if the difference  
2 affected execution of the speculative thread involves determining if the  
3 speculative thread accessed the predicted result.

1 4. (Original) The method of claim 1, wherein determining if the difference  
2 affected execution of the speculative thread involves determining if the predicted  
3 result differs from the result generated by the head thread.

1 5. (Original) The method of claim 1, wherein generating the predicted  
2 result involves looking up a value based upon a program counter for the program.

1 6. (Original) The method of claim 5, wherein generating the predicted  
2 result involves additionally looking up the value based upon at least one  
3 previously generated value for the result.

A<sup>1</sup>

1           7. (Original) The method of claim 5, wherein generating the predicted  
2 result involves performing a function on the value.

1           8. (Original) The method of claim 1, wherein executing the section of code  
2 involves performing one of:  
3           a method invocation to execute the section of code;  
4           a function call to execute the section of code; and  
5           a procedure call to execute the section of code.

1           9. (Original) The method of claim 1, wherein the section of code is a body  
2 of a loop in the program, and the result is a loop carried dependency for the loop.

1           10. (Original) The method of claim 1, wherein during a write operation to  
2 a memory element by the head thread, the method further comprises:  
3           performing the write operation to a primary version of the memory  
4 element;  
5           checking status information associated with the memory element to  
6 determine if the memory element has been read by the speculative thread;  
7           if the memory element has been read by the speculative thread, causing the  
8 speculative thread to roll back so that the speculative thread can read a result of  
9 the write operation; and  
10          if the memory element has not been read by the speculative thread,  
11 performing the write operation to a space-time dimensioned version of the  
12 memory element if the space-time dimensioned version exists.

1           11. (Original) The method of claim 10, wherein performing the join  
2 operation involves merging the space-time dimensioned version of the memory

AI  
3 element into the primary version of the memory element and discarding the space-  
4 time dimensioned version of the memory element.

1 12. (Currently amended) An apparatus that facilitates predicting a result  
2 produced by a section of code in order to support speculative program execution,  
3 the section of code including a plurality of program instructions, the apparatus  
4 comprising:

5 a head thread that is configured to execute the section of code within a  
6 program, wherein executing the section of code produces the result;

7 a prediction mechanism that is configured to generate a predicted result to  
8 be used in place of the result before the head thread produces the result;

9 a speculative thread that is configured to speculatively execute subsequent  
10 code within the program using the predicted result, wherein the subsequent code  
11 follows the section of code in an execution stream of the program, and wherein  
12 speculatively executing the subsequent code involves performing one of:

13 a speculative method invocation to speculatively execute  
14 the subsequent code,

15 a speculative function call to speculatively execute the  
16 subsequent code, and

17 a speculative procedure call to speculatively execute the  
18 subsequent code;

19 wherein the head thread and all speculative threads execute instructions  
20 from separate instruction caches; and

21 a determination mechanism that is configured to determine if a difference  
22 between the predicted result and the result generated by the head thread affected  
23 execution of the speculative thread; and

A1  
24 a joining mechanism that is configured to merge state associated with the  
25 speculative thread with state associated with the head thread if the difference did  
26 not affect execution of the speculative thread;  
27 wherein if the difference affected execution of the speculative thread, the  
28 apparatus is configured to execute the subsequent code again using the result  
29 generated by the head thread.

1 13. (Original) The apparatus of claim 12, wherein while executing the  
2 subsequent code again, the apparatus is configured to perform a rollback operation  
3 for the speculative thread to undo actions performed by the speculative thread.

1 14. (Original) The apparatus of claim 12, wherein the determination  
2 mechanism is configured to determine if the speculative thread accessed the  
3 predicted result.

1 15. (Original) The apparatus of claim 12, wherein the determination  
2 mechanism is configured to determine if the predicted result differs from the  
3 result generated by the head thread.

1 16. (Original) The apparatus of claim 12, wherein the prediction  
2 mechanism is configured to generate the predicted result by looking up a value  
3 based upon a program counter for the program.

1 17. (Original) The apparatus of claim 16, wherein the prediction  
2 mechanism is configured to generate the predicted result by additionally looking  
3 up the value based upon at least one previously generated value for the result.

A1  
1 18. (Original) The apparatus of claim 16, wherein the prediction  
2 mechanism is configured to generate the predicted result by performing a function  
3 on the value.

1 19. (Original) The apparatus of claim 12, wherein the section of code  
2 includes one of, a method, a function and a procedure.

1 20. (Original) The apparatus of claim 12, wherein the section of code is a  
2 body of a loop in the program, and the result is a loop carried dependency for the  
3 loop.

1 21. (Original) The apparatus of claim 12, further comprising a mechanism  
2 that performs write operations for the head thread, the mechanism being  
3 configured to:

4 perform a write operation to a primary version of a memory element;  
5 check status information associated with the memory element to determine  
6 if the memory element has been read by the speculative thread;

7 cause the speculative thread to roll back so that the speculative thread can  
8 read a result of the write operation if the memory element has been read by the  
9 speculative thread; and

10 perform the write operation to a space-time dimensioned version of the  
11 memory element if the space-time dimensioned version exists and if the memory  
12 element has not been read by the speculative thread.

1 22. (Original) The apparatus of claim 21, wherein the joining mechanism  
2 is configured to:

3 merge the space-time dimensioned version of the memory element into the  
4 primary version of the memory element; and to

5 discard the space-time dimensioned version of the memory element.

1 23. (Currently amended) A computer-readable storage medium storing  
2 instructions that when executed by a computer cause the computer to perform a  
3 method that predicts a result produced by a section of code in order to support  
4 speculative program execution, the section of code including a plurality of  
5 program instructions, the method comprising:

6 executing the section of code within a program using a head thread,  
7 wherein executing the section of code produces the result;  
8 before the head thread produces the result, generating a predicted result to  
9 be used in place of the result;

10 allowing ~~the speculative~~ a speculative thread to speculatively execute  
11 subsequent code within the program using the predicted result, wherein the  
12 subsequent code follows the section of code in an execution stream of the  
13 program, and wherein speculatively executing the subsequent code involves  
14 performing one of:

15 a speculative method invocation to speculatively execute  
16 the subsequent code,

17 a speculative function call to speculatively execute the  
18 subsequent code, and

19 a speculative procedure call to speculatively execute the  
20 subsequent code;

21 wherein the head thread and all speculative threads execute instructions  
22 from separate instruction caches; and

23 after the head thread finishes executing the section of code, determining if  
24 a difference between the predicted result and the result generated by the head  
25 thread affected execution of the speculative thread;

A1  
26 if the difference affected execution of the speculative thread, executing the  
27 subsequent code again using the result generated by the head thread; and  
28 if the difference did not affect execution of the speculative thread,  
29 performing a join operation to merge state associated with the speculative thread  
30 with state associated with the head thread.

1 24. (Original) The computer-readable storage medium of claim 23,  
2 wherein executing the subsequent code again involves performing a rollback  
3 operation for the speculative thread to undo actions performed by the speculative  
4 thread.